第3章

システムLSI上にパケット転送のネットワーク・システムを構築

―― 非同期オンチップ・ネットワークの設計手法

W. J. Bainbridge, A. Bardsley, R. W. McGuffin

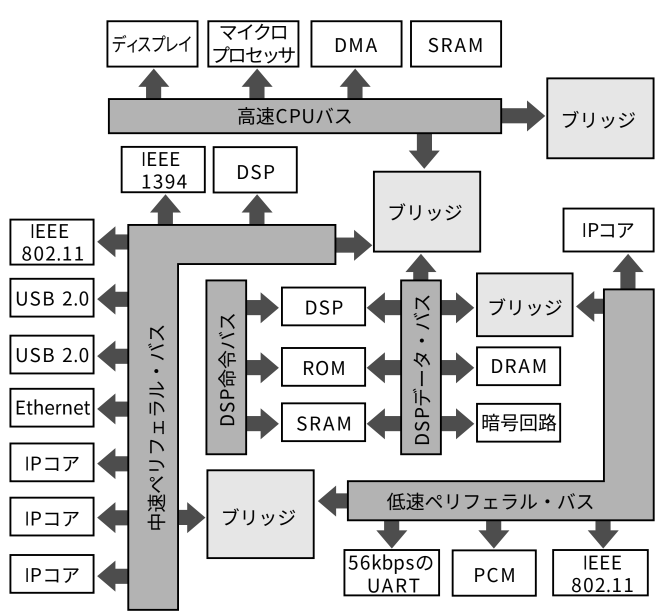
CHAINオンチップ・ネットワークは，階層型バス接続より優れているネットワーク接続の利点と，セルフタイムド手法による設計（いわゆる非同期設計方式）が備えるタイミング収束や低消費電力の利点を合わせ持つ手法である．この手法を導入すると，最新のプロセスで製造されたLSIが抱えるいくつかの配線関連の問題，すなわち複雑な配線構造や消費電力管理，タイミング収束などの問題を柔軟に，そしてクロックに依存しない形で解消することができる．ここでは小規模な試作チップを例に，CHAINアーキテクチャの基本的な考えかたを解説する．また，セルフタイムド手法によるオンチップ・ネットワークの設計を支援するツールやライブラリも紹介する．（筆者）

システムが複雑になり，また，CMOS LSIの集積度が上がった結果，素子遅延（ゲート遅延）よりも配線遅延の影響のほうが大きくなってきています．これに伴って，システムLSIに集積する回路ブロック（コンポーネント）の間を適切に接続することがだんだん難しくなってきています．

現在，LSI設計者は，場当たり的に階層化されたクロック同期のバスや，それらの間をつなぐブリッジを利用しています（**図1**）．バス・クロック周波数は，そのバスにつながる周辺回路（デバイス）の通信要求（伝送性能など）に基づいて決定されます．ただし，実際に得られるバスの動作速度は，物理的なバス配線の長さや，レイアウトの最適化にどれだけ多くの時間をかけられたかに左右されます．

階層型バス接続をベースとするLSIは，徐々に設計することが困難になってきています．電話システムが回線交換からパケット交換へと移行したように，回路ブロック間の接続は現行の階層型バス接続から，この難題を解消できるネットワーク技術を使った接続へと移行することでしょう．本稿では，セルフタイムド手法を使ってオンチップ・ネットワークを実装することにより，設計を複雑にしているタイミング収束や消費電力管理の課題を解決できることを説明します．

セルフタイムド手法のオンチップ・ネットワーク（例えば筆者らが提供するCHAINオンチップ・ネットワーク）を中核としてシステムLSIを構築すると，それぞれ異なる周波数で動作するIPコアどうしを（それが同期設計によるものであっても，セルフタイムド手法で設計されたものであっても）そのまま接続できます．その結果，ITRS（International Technology Roadmap for Semiconductors）のロードマッ



PCM：pulse code modulation
UART：universal asynchronous receiver-transmitter

図1　階層型バス接続の例

現在のLSI設計では，階層化されたクロック同期のバスやそれらの間をつなぐブリッジが利用されている．

3

プに示されているようなGALS（globally-asynchronous locally-synchronous；大域的に見ると非同期，局所的に見ると同期）構成のLSIが得られます．

● バスの代わりにリンクとスイッチで通信網を構成

CHAIN技術(4)は，システムLSI上の回路ブロックの間を接続する一つの手法です．この手法では，低コストの（つまり，チップ面積にほとんど影響を与えない）スイッチと狭帯域のリンク（チャネル）によって構成されるネットワークを利用します．このネットワークは，信号遅延の影響を受けないセルフタイムド手法の通信プロトコルに従って動作します．

伝送性能を引き上げるため，リンクやスイッチを2重，3重に束ねて使用することもできます．また，クライアントとなるデバイス（回路ブロック）との間に仮想的なコネクションを実現するため，厳密に定義されたパケット・フォーマットと組み合わせて利用することも可能です．

図2にCHAIN技術によるリンクの例を示します．一つのリンクは五つの送信信号と一つのアクノリッジ信号から構成されます．これらの信号を使ってセルフタイムド方式のフロー制御を実現します．また，このリンクは，データ符号化方式としてワンホット・エンコーディングを利用して，2ビットのデータや制御マーカ，eop（end of packet）マーカを送信します．データを送った後，RTZ（return to zero）プロトコルに従ってアイドル状態に戻ります．そのため，2ビットの伝送に対して全部で4回（データ信号について2回，アクノリッジ信号について2回）の信号変化が発生します（いわゆる4相ハンドシェーキング）．

● リピータがパイプライン・ラッチに置き換わる

最先端のCMOSプロセスを利用して製造されたLSIでは，長い配線において重大な信号遅延が生じます（信号の伝播遅延時間は配線長に比例するため）．これは，信号波形を元に戻すために，複数のリピータの挿入が必要になることを意味します．

CHAINオンチップ・ネットワークでは，これらのリピータがパイプライン・ラッチに置き換わります．これによって，リンクのサイクル・レート（動作周波数）は上がりますが，レイテンシが増えることはほとんどありません．この種の置き換えでは，レイテンシにほとんど影響を与えることなくパイプラインの深さを変更できます．これがセルフタイムド手法の利点の一つになっています（これに対して同期設計方式では，パイプラインの段数を増やすと，かならずレイテンシが増加する）．

結果として，ファブリック（チップ全体）の性能を制限しているグローバル・クロックの問題がなくなり，ネットワークの中のリンクを個別に調整（最適化）できるようになります．**図3**にCHAINパイプライン・ラッチの回路例を示します．この回路の中のキーとなる部分は，データパス・ラッチ（実際には，トランスペアレント・ラッチに似たMuller Cエレメント(6)）とORゲートです．ORゲートは有効なラッチ出力を検出し，アクノリッジ信号を生成します．

CHAINリンクのリンク長（配線長）に対するサイクル・レートの変動に関してシミュレーションした結果を**図4**に示します．モデルとしては，台湾TSMC（Taiwan Semiconductor Manufacturing Co.）が提供している180nm CMOSプロセス（配線はAl）のSPICEモデル（*RC*モデル）

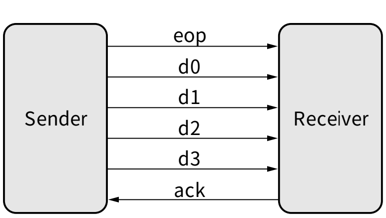
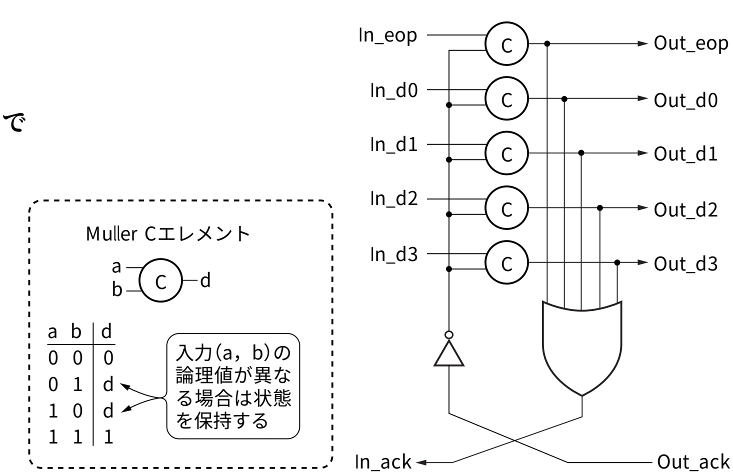


図2 CHAINリンク
一つのリンクは五つの送信信号（eop，d0，d1，d2，d3）と一つのアクノリッジ信号（ack）から構成されている．



a	b	d
0	0	0
0	1	d
1	0	d
1	1	1

図3 CHAINパイプライン・ラッチ
この回路の中のキーとなる部分は，データパス・ラッチとORゲート．ORゲートは有効なラッチ出力を検出し，アクノリッジ信号を生成する．

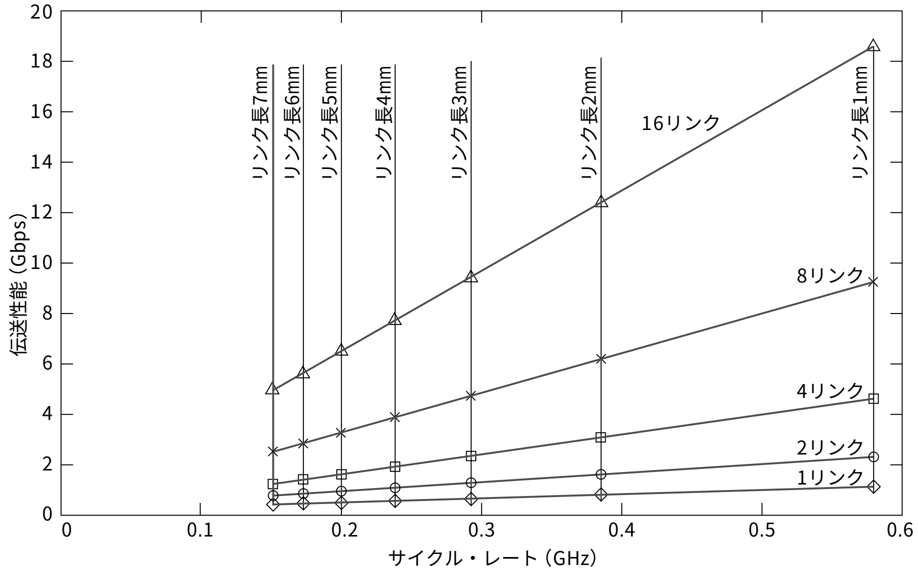


図4
CHAIN リンクのサイクル・レートとリンク長
リンク長が2mmの場合，サイクル・レートは最大380MHz，伝送性能は1リンク当たり約0.75 Gbps．さらに高い伝送性能が必要な場合は，複数のリンクを束ねて使用することもできる．

を使用しました．リンク長が2mmの場合，サイクル・レートは最大380MHz，伝送性能（スループット）は1リンク当たり約0.75Gbpsであることがわかります．さらに高い伝送性能が必要な場合には，データパス（伝送路）のバンド幅を広げるために，複数のリンクを束ねて使用します．

なお，ここで使用したリピータは，駆動力がx2の標準セルです．駆動力の高いセルを使用すれば，上記のサイクル・レートは500MHzまで上がります．

● データ転送にパケット交換方式を導入可能

図3のパイプライン・ラッチ（リピータ）を拡張して[4]，スイッチ・ネットワークの基本モジュールを構成することも可能です．CHAINオンチップ・ネットワークは，分散型スイッチング・ファブリックの考えかたを採用しています．データはネットワーク・ゲートウェイの間でパケットの形で受け渡されます．パケットの先頭には経路情報が納められており，パケットはこれに基づいて受信側のゲートウェイに転送されます．一方，パケットの末尾にはeopマーカが存在します．eopマーカによってパケットが通過したことを検出すると，即座にその経路はクリアダウン（消去）されます．

後述する試作チップ「Buxton[3]」のオンチップ・ネットワークでは，以下のようなパケット・フォーマットを採用しました．

- **パケットの先頭は経路シンボル**
- **8ビットのパケット・ヘッダには，トランザクションのシーケンス・タグとイニシエータ識別子が含まれる**
- **8ビットのバス・メッセージ・ヘッダでバス制御信号を伝送する．また，ペイロードで伝送されるバス転送データのタイプが示される**
- **ペイロードは16ビット単位に分割されている**
- **パケットの最後はeopマーカ**

このオンチップ・ネットワーク内の接続において，複数のリンクを束ねて使用する場合，経路シンボルとeopシンボルについては全リンクでコピーされ，パケット本体（メッセージ）については各リンクに振り分けて送信されます．試作チップで採用したこの方式は，あくまでもリンクを束ねて使用する方法の一例であり，このほかにもいくつかの方式が考えられます．例えば，経路シンボルも各リンクに振り分ける方法や，特定のリンクを使って経路情報やeop情報を転送し，これ以外のリンクとの間にマスタ・スレーブの関係を作るような方法です．

● ゲートウェイとインターフェース・アダプタを用意

ネットワーク・ファブリックの先にはネットワーク・ゲートウェイがあり，これはスイッチ・ネットワーク全体のメッセージ交換に必要なさまざまな制御，およびデータの分解・組み立てを実行します．例えば，以下のような機能を備えています．

- **パラレル・データを狭帯域のコネクションを使って伝送するためのシリアライザ/デシリアライザ**
- **送信側から受信側へパケットを導くために必要となる経**

路シンボルの決定と，パケットの先頭への挿入（送信）

- **パケットとペイロードのタイプを示すメッセージ・ヘッダの組み立て**
- **メッセージの末尾に位置し，ネットワーク上のパケット転送経路をクリアダウンするeopシンボルの送信**
- **ネットワーク・ファブリックの輻輳（ふくそう）やゲートウェイ受信バッファの混雑を最小限に抑えるためのフロー制御**

ゲートウェイは，通常，ネットワークに対して二つのポートを備えています．一つはパケットの送信用で，もう一つは受信用です．ネットワーク・ファブリックのトポロジ（接続構造）に従って，こうしたゲートウェイは，リンクでつながれたクライアントとなるIPコアやバスを含むIPサブシステムに対して，イニシエータ（データ転送を要求する側）になったり，ターゲット（データを提供する側）になったり，あるいは両方の役目を果たしたりします．試作チップで採用されたイニシエータ・ゲートウェイの内部構成を**図5**に示します．

筆者ら（英国Silistix社）が開発したネットワーク・ゲートウェイは，セルフタイムド手法のプロトコル（CGP：CHAIN Gateway Protocol）に対応した独自のソケット・フォーマットを採用して通信しています．また，今日の業界標準バス（AMBA[1]やOCP，CoreConnect[2]）に対応した既設計のIPコアを再利用できるように，CGPソケットと従来のプロトコルの間のマッピングを行うクライアント・アダプタのライブラリを開発しました．このアダプタは，バスの概念をネットワークの機能にマッピングします．加えて，バス・トランザクションの転送要求処理や，セルフタイムドなCGPソケットと同期設計方式のIPコア（従来のバス・プロトコル）の間の変換処理も行います．

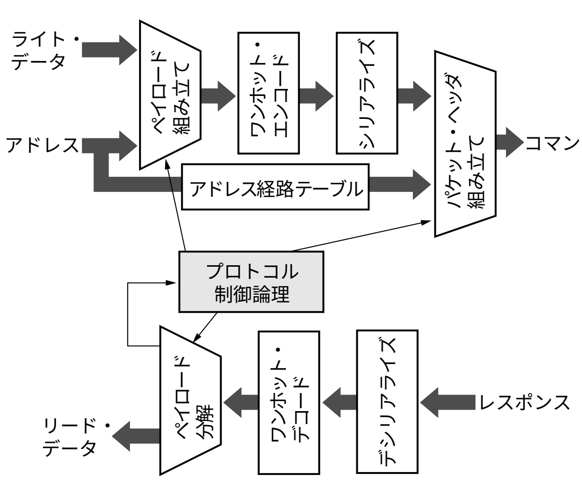


図5　イニシエータ・ゲートウェイ

ゲートウェイは，リンクでつながれたクライアントとなるIPコアやバスを含むIPサブシステムに対して，イニシエータ（データ転送を要求する側）になったり，ターゲット（データを提供する側）になったりする．ここではイニシエータの構造を示している．

● 大学がスマート・カード用LSIを試作

本稿で紹介するオンチップ・ネットワーク技術の多くは，英国University of Manchesterが開発したスマート・カード用の試作チップ[3]に実装されています．このチップは，TSMCの180nmのCMOSプロセスで製造されました．

このLSIのオンチップ・ネットワークは，三つのイニシエータ・ゲートウェイ，六つのターゲット・ゲートウェイ，そして外部からオンチップ・ネットワークに直接アクセスできるネットワーク・デバッグ・ポートを搭載しています．また，このネットワークは二つの独立したセグメントからなり，第1のセグメントではイニシエータからターゲットへコマンドを送信し，第2のセグメントではイニシエータへ応答を返します．

本試作チップは，イニシエータとして2個の（セルフタイムド手法で設計した）マイクロプロセッサとテスト・インターフェースを，ターゲットとして各種メモリとUART（universal asynchronous receiver transmitter），および実験用ユニットを備えています．プロセッサについては，CGPソケットと接続するために，あらためてインターフェースに手を加える必要はほとんどありませんでした．一方，これ以外のデバイスについては，小規模なアダプタ・ラッパ回路を付加する必要がありました．

● 4個のリンクとスイッチ部品を組み込む

スマート・カードに必要な処理性能はたかだか10MIPS程度なので，このチップに複雑な高性能ネットワークは不要なのですが，ここではCHAINが備える以下の三つの拡張性を利用しました．

- **リンク長**── 性能要求が高くなかったため，1リンク当たり約0.75Gbpsの伝送性能が得られる約2mmのリンクを使用した．
- **束ねたリンクの数**── 二つのリンクを利用して，合計1.5Gbpsの伝送性能を得た．これが，約3,100万メッセージ/sのメッセージ転送性能（48ビットの読み取りコマンドや読み取り応答メッセージの転送），およびファブリックに固有の32nsのレイテンシ（経路設定時間など）の上

限値を規定する．

- **ネットワーク・トポロジ**── 二つのプロセッサが（一般的な状況において）同時にイネーブルされることはなく，メモリ・システムのパイプライン処理にも対応していないので，それらの同時並行動作が要求されることはほとんどない．そのため，共有バスに似たシンプルな集中・分散型のトポロジで十分だった．

このファブリックは，コマンドを送る二つの送信リンク，応答を返す二つの受信リンク，そして，**図6**のようなスイッチ部品から構成されています．

● タイミング検証の作業が簡素化される

チップの最終的なレイアウトを**図7**に示します．CHAINリンクとファブリック・スイッチの場所もいっしょに示します．LSIのフロアプランに関する制約は，各リンク長を2mm未満にし，かつ，だいたいその長さをそろえるというものでした．これは難しい制約ではありませんでした．

機能の妥当性の検証，および演算ブロックのタイミング動作の確認・修正を行うことを目的に，チップ全体のシミュレーションを行いました．このシミュレーションでは，レイアウト情報から（寄生容量/抵抗抽出ツールで）抽出した二つのプロセッサのトランジスタ・レベルのネットリスト（*RC*モデル），メモリの機能モデル，CHAINネットワークのトランジスタ・レベルのネットリスト（ツールで抽出したものではない）を使用しました．

チップを製造する前に，CHAINネットワーク・ファブリックの寄生容量/抵抗の抽出は行いませんでした．というのは，セルフタイムド手法による設計では（随時，修正を加えることで）適切なタイミング動作が保証されており，加えて，このオンチップ・ネットワークには要求されるシステム性能に対してかなりのマージンを持たせてあったためです．

先に述べたように，CHAINネットワークでは，リンク長やリンク数を自由に選択できます．つまり，要求されるネットワーク性能に合わせてリンクの構成をきめ細かく指定できるのです．また，マージンを確保しやすいCHAINネットワークを中核としてシステムLSIを構築すれば，システム・レベルの接続（グローバル配線など）のタイミング検証の作業をかなり簡素化できます．

高性能なシステムLSIを開発する場合，一般に，機能は適切に設計されるのですが，これとは別に，オンチップ・ネットワークの性能を確認するために，なんらかのシミュレーションを行うことが望ましいでしょう．

オンチップ・ネットワークの設計とこの試作チップに必要な回路部品の作成はいずれも人手で行いました．それ以外の工程（ゲート・レベル・シミュレーション，静的タイミング解析，レイアウト，寄生容量/抵抗抽出，トランジスタ・レベル・シミュレーションなど）では，米国Synopsys社や米国Cadence Design Systems社のツール

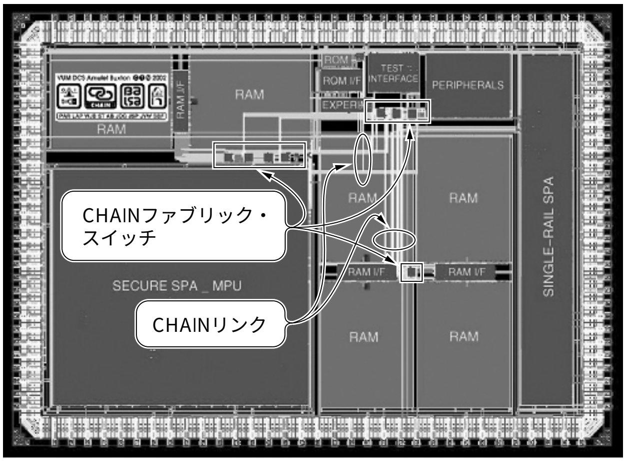


図7　スマート・カード用LSI「Buxton」
英国University of Manchesterが開発した試作チップ．TSMCの180nmのCMOSプロセスで製造された．

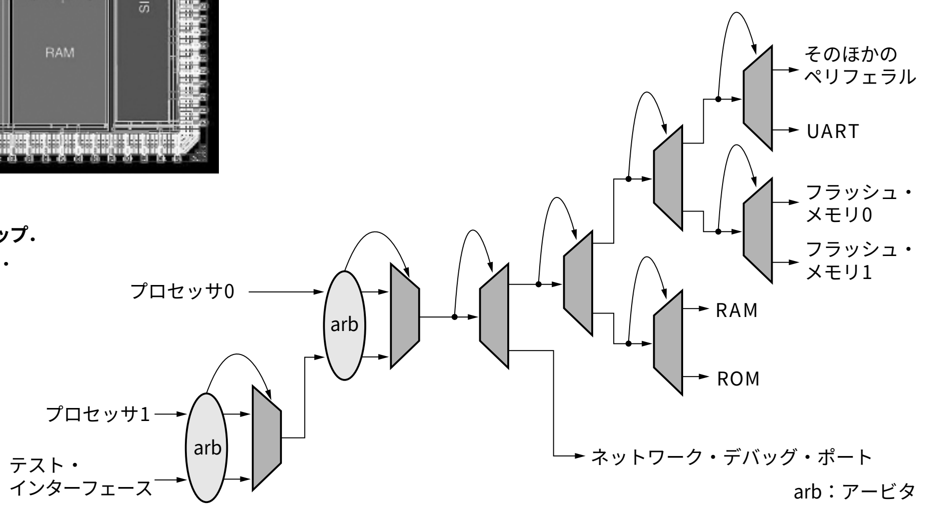


図6
ネットワーク・ファブリックの構成
スイッチ部品の構成を示している．

を使用しました．セル・ライブラリには，標準的なセルのほか，少数のセルフタイムド手法専用のセル（ミューテックス[7]，Muller Cエレメント[6]など）が組み込まれています．これらの専用セルは，従来の同期設計用の標準セルを組み合わせて構成することもできるのですが，今回は性能や面積を最適化するために，カスタム設計を行いました．

● オンチップ・ネットワークは500MHz以上で動作

ネットワークの中のゲートウェイ（ターゲット）の一つには，デバッグしたり，ネットワーク性能を測定したりするための回路を追加しました．このチップのデバッグおよびCHAINネットワークの機能の詳細については，参考文献(5)を参照してください．

測定の結果，CHAINファブリックが実際に500MHz以上のサイクル・レートで動作することが確認されました．これは，**図4**に示したように，シミュレーションで予測されていたとおりの性能です（チップ上のもっとも長いリンクは約1.5mm）．先に述べたように，すべてのリンクについてドライバの駆動能力が不足する状況に遭遇しました．これについては，適切なバッファの挿入を行えば，さらに高速に動作するはずです．

試作チップの消費電力は測定しませんでした．これは，比較対象となる等価な同期設計のLSIが存在しなかったためです．また，電力消費は回路の動作にも左右されますが，システム・ソフトウェアの影響も受けます．

● オンチップ・ネットワーク自動生成ツールを開発

ここで紹介したスマート・カード用LSIのオンチップ・ネットワークは，セルフタイムド手法の専門家が回路図入力によって設計しました．また，アーキテクチャについては，限定された範囲で複数案の検討を行いました．

ただし，これまで同期設計を行ってきたLSI設計者がCHAINネットワークの技術を商用LSIの開発に適用しようとした場合，上述のような手法で十分とは言えません．筆者らが所属するSilistix社は，University of Manchesterを通じて融資を受けているベンチャ企業です．この会社は，候補となるアーキテクチャのトレードオフ評価を自動化し，選択されたオンチップ・ネットワークを自動生成（コンパイル）するためのソフトウェア・ツール「CHAINworks」とライブラリの開発を行っています．

筆者らの開発したツールを利用すると，これまで同期設計を行ってきたシステムLSIの開発者でも（例えば，非同期設計の複雑なノウハウを理解していなくても），ネットワーク接続の利点とセルフタイムド手法の利点を比較的容易に享受することができます．以下では，本ツールの構成について簡単に述べます．

1）入力データ

本ツールの入力は，オンチップ・ネットワークのポートのSystemVerilog構造記述（ネットリスト），あるいはブロック図データ（例えば，米国Mentor Graphics社の「Design Architect」やCadence社の「Composer」で作成したもの）です．これらのデータの中のバス・モジュールは，クライアント（IPコア）につながるポートを含んでいます．それぞれのポートにはアダプタ・ユニットの回路が用意されており，業界標準のバス・インターフェース（AMBAなど）を利用して回路ブロックを接続したい場合にはこのアダプタ・ユニットを使用します．

オンチップ・ネットワークの接続要件（例えば，異なるクライアントの間の伝送性能やクライアントのトラフィック・プロファイル，インターフェースのデータ・バス幅，クライアントのクロック周波数など）が明確になるように，ネットワーク・モジュールとアダプタにはいくつかの制約が設けられています．

2）アーキテクチャの評価

本ツールでは，ユーザが与えるインターフェース要求や接続上の制約に基づいて，さまざまなアーキテクチャのトレードオフ評価を対話的に行えます．例えば，トポロジやパイプラインの深さ，リンク幅を変更しながら，オンチップ・ネットワークの占有面積や消費電力，性能，検証に要する期間などを比較・検討します．また，トラフィックがパターン化されたビヘイビア・バス・モデルによるシミュレーションを行い，その結果をトレードオフ評価の工程に反映させます．こうした作業は，設計の最初の段階で行えます．そのため，後工程におけるタイミングに関する設計修正の回数が減り，全体の設計効率が上がります．

3）設計フローと出力結果

本ツールは，既存の設計フローにそのまま組み込むことができます．設計フローの中の本ツールの位置づけを**図8**に示します．この図は，ほかのツールとの関係やタイミング収束のループとの関係も示しています．

本ツールの入力データは，現状では，制約コメント付きのSystemVerilog構造記述です．一方，出力データは，

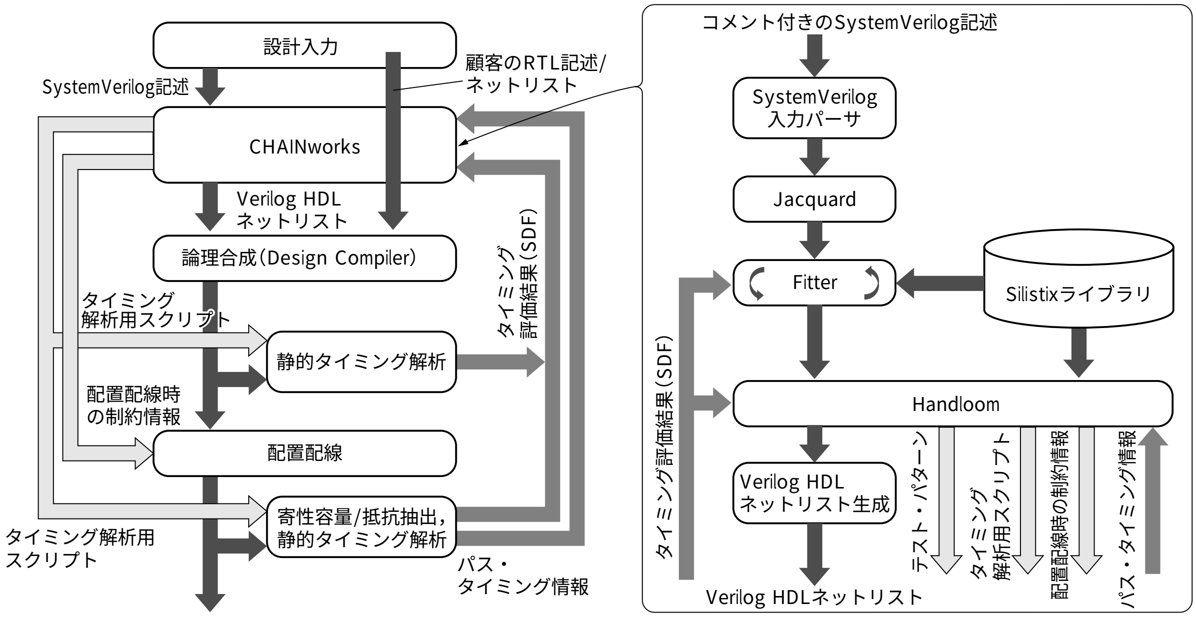


図8　設計フローの中のCHAINworksの位置づけ
CHAINworksは，既存の設計フローにそのまま組み込むことができる．

オンチップ・ネットワークのVerilog HDLネットリスト(Verilog 95に準拠)，およびさまざまな工程で利用される補助データです．例えば，以下のデータが出力されます．

- **システム・レベル・シミュレーションに利用されるVerilog HDLとSystemCのビヘイビア・モデル**
- **ネットワーク部品の配置配線やタイミング解析に利用されるタイミング制約データ(SDFフォーマット)**
- **LSIテストに使用されるオンチップ・ネットワークのテスト・パターン**

4）ツールの内部処理

本ツールは，三つの主要なコンポーネント「Jacquard」，「Fitter」，「Handloom」から構成されています(**図8**の右側を参照)．本ツールの合成機能は，小規模なハードウェア・テンプレートを組み合わせて，大規模なモジュールを構築(コンパイル)します．ライブラリには，基本的な回路素子，パイプライン・ステージ，スイッチ，シリアライザからネットワーク・ゲートウェイまで，多様なテンプレートが格納されています．また，これらの部品は，すべてセルフタイムド手法に基づいて設計されており，データ幅や符号化方式，動作速度などをパラメータで指定できるようになっています．

使用する部品の選定とこれらの接続はFitterが実行します．制約に関係する部分については，Jacquardを利用してユーザが手を加えられます．満足のいく設計が得られたら，バックエンドのHandloomがテクノロジ・マッピングを行い，最終的な構造記述(ネットリスト)を生成します．

参考・引用*文献

(1) ARM； *AMBA, Advanced Microcontroller Bus Architecture Specification, Rev 2.0*，May 1999.
(2) IBM； *CoreConnect Bus Architecture, product brief*，http://www.chips.ibm.com/news/1999/990923/pdf/corecon128_pb.pdf
(3) L. A. Plana，P. A. Riocreux，W. J. Bainbridge，A. Bardsley，S. Temple，J. D. Garside and Z. C. Yu；“SPA - a Secure Amulet Core for Smartcard Applications”，*Microprocessors and Microsystems*，pp. 431-446，vol. 27，no. 9，Oct. 2003.
(4) W. J. Bainbridge and S. Furber；“Chain: a Delay-Insensitive Chip Area Interconnect”，*IEEE Micro*，pp. 16-23，vol. 22，no.5，Sep./Oct. 2002.
(5) W. J. Bainbridge，L. A. Plana and S. B. Furber；The Design and Test of a Smartcard Chip using a CHAIN self-timed network-on-chip，*Proc. DATE 04*(*Paris*)，pp.274，vol. 3，Feb. 2004.
(6) M. Shams，J. C. Ebergen and M. I. Elmasry；“Modelling and comparing CMOS implementations of the C-element”，*IEEE Transactions on VLSI Systems*，pp. 563-567，vol. 6，no. 4，Dec. 1998.
(7) C. L. Seitz，“System timing” in C. A. Mead and L. A. Conway；*Introduction to VLSI Systems*，chapter 7，Addison-Wesley，1980.

W. J. Bainbridge，A. Bardsley，R. W. McGuffin
英国Silistix社

3